TALN 2010, Montréal, 19–23 juillet 2010

Convertir des dérivations TAG en dépendances

Éric Villemonte de La Clergerie
Alpage, INRIA / Univ. Paris 7, 30 rue du Château-des-rentiers, 75013 Paris
eric.de_la_clergerie@inria.fr

Résumé. Les structures de dépendances syntaxiques sont importantes et bien adaptées comme point de départ de diverses applications. Dans le cadre de l'analyseur TAG FRMG, nous présentons les détails d'un processus de conversion de forêts partagées de dérivations en forêts partagées de dépendances. Des éléments d'information sont fournis sur un algorithme de désambiguisation sur ces forêts de dépendances.

Abstract. Syntactic dependency structures are important and adequate as starting point for various NLP applications. In the context of the French TAG FRMG parser, we present the details of a conversion process from shared derivation forests into shared dependency forests. Some information are also provided about a disambiguisation algorithm for these dependency forests.

Mots-clés : dépendances, analyse syntaxique,TAG, forêt partagée.

Keywords: dependencies, parsing,TAG,shared forest.

1 Introduction

Les sorties d'analyse syntaxique sont potentiellement très utiles comme point de départ pour diverses applications, en particulier d'extraction d'information (Éric de La Clergerie *et al.*, 2009) ou d'acquisition de connaissances, par exemple en tant que triplets de dépendances, de chemins entre mots ou de configurations prédicats-arguments. Néanmoins, ces sorties sont parfois complexes et pas directement exploitables, nécessitant par exemple de convertir une sortie « concrète » en une sortie plus abstraite. Cette tâche, un peu ingrate, est rarement documentée, alors qu'elle nécessite une définition claire des sorties traitées.

Nous nous intéressons aux traitements des sorties de l'analyseur syntaxique FRMG. Résultant de la compilation par le système DYALOG (Villemonte de la Clergerie, 2002) d'une grammaire TAG du français à large couverture (Thomasset & Villemonte de La Clergerie, 2005), FRMG retourne, sous forme de forêt partagée, l'ensemble (potentiellement exponentiel) des dérivations TAG couvrant complètement ou partiellement une phrase. En pratique, même si les arbres de dérivations sont plus abstraits que les arbres d'analyses et plus proches d'un niveau sémantique (Gardent & Kallmeyer, 2003), ils ne sont pas forcément adaptés pour de traitements ultérieurs, car mettant essentiellement en relation des arbres TAG de la grammaire sous-jacente. Heureusement, le caractère fortement lexicalisé des arbres TAG permet d'explorer la conversion des arbres de dérivations en arbres de dépendances. Bien que ce fait soit relativement bien connu (Candito & Kahane, 1998; Joshi & Rambow, 2003), nous précisons ici quelques points concrets de cette conversion. L'originalité de notre approche est également de généraliser la conversion de manière à obtenir des forêts partagées de dépendances et de proposer une représentation de ces forêts.

Les forêts ambiguës sont parfois intéressantes, par exemple pour des expériences d'acquisition (Fernandez *et al.*, 2007). Néanmoins, les applications s'appuyant sur l'analyse syntaxique requièrent généralement une structure non-ambiguë de dépendances, d'où la brève présentation d'un algorithme de désambiguisation.

2 Quelques rappels sur les TAG

Les grammaires d'arbres adjoints [TAG (Joshi, 1987)] s'appuient sur la combinaison d'arbres élémentaires par les opérations de substitution ou d'adjonction pour construire des arbres d'analyse. En particulier, l'opération d'adjonction, qui ouvre un noeud interne pour y greffer le contenu d'un arbre auxiliaire, permet la gestion de dépendances à longue distance dans les phrases. Par ailleurs, l'utilisation d'arbres offre la notion de *domaine de localité étendu* et permet plus directement la lexicalisation des grammaires. Un arbre peut en effet être *ancré* par une tête lexicale, les autres noeuds (éventuellement lexicalisés) formant son domaine de localité. Typiquement, un arbre ancré par un verbe contient un noeud pour chaque argument sous-catégorisé par celui-ci.

La figure 1 montre les arbres élémentaires permettant l'analyse de « *Yves donne un joli livre à Sabine* », avec, en particulier, un arbre verbal ancré par *donne* sous-catégorisant pour un verbe ditransitif et un arbre ancré par *joli* venant modifier *livre* par adjonction. L'arbre d'analyse obtenu est donné par la figure 2a. Mais, classiquement, l'information utile est fournie par l'arbre de dérivation de la figure 2b qui indique l'ensemble des opérations de dérivations ayant pris place, en précisant, pour chaque opération, l'arbre et le noeud d'origine et l'arbre ajouté (par substitution ou adjonction). Ainsi, l'arbre exemple stipule qu'une substitution prend place sur le noeud NP_1 de l'arbre `tn1pn2` ancré par *donne*, introduisant l'arbre `np` ancré par *Yves*. L'arbre de dérivation est pivot dans le sens où il permet la reconstruction de l'arbre d'analyse, tout en se rapprochant d'un niveau de représentation plus sémantique (Candito & Kahane, 1998; Gardent & Kallmeyer, 2003). Ce modèle de base est enrichi dans le cadre de FRMG.

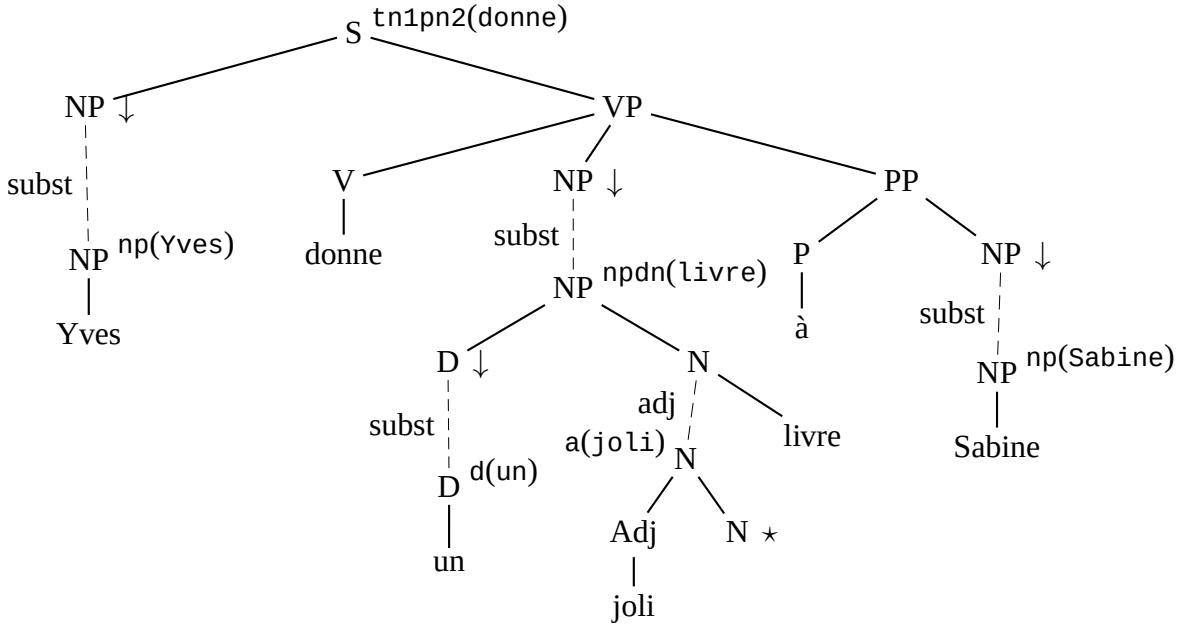
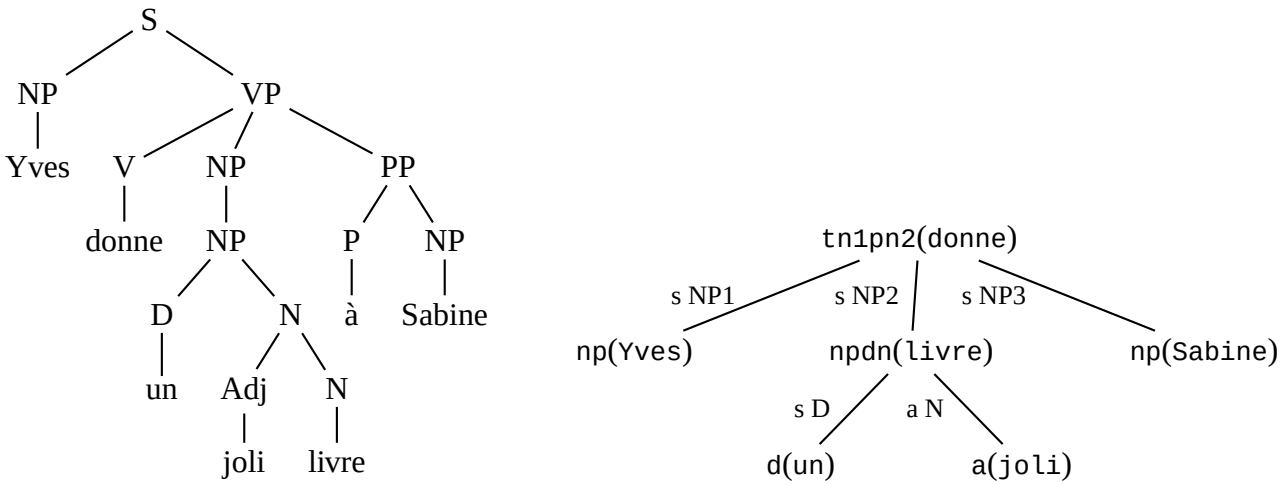


FIGURE 1 – Arbres élémentaires pour *Yves donne un joli livre à Sabine*



(a) Arbre d'analyse (b) Arbre de dérivation

FIGURE 2 – Arbres d'analyse et de dérivation

3 FRMG

L'analyseur FRMG est issu de la compilation par le système DYALOG d'une grammaire TAG large couverture du français générée par une méta-grammaire (Thomasset & Villemonte de La Clergerie, 2005). Une de ses principales caractéristique est la gestion d'un petit nombre d'arbres (environ 200) grâce à l'emploi d'opérateurs de factorisation sur les noeuds (disjonction, répétition, optionalité, gardes, entrelacement). Bien que la factorisation ne change ni le pouvoir expressif des TAG ni leur complexité, un arbre factorisé peut représenter un nombre exponentiel d'arbres TAG classiques (Villemonte de la Clergerie, 2010).

Au plus un noeud ancre est distingué par arbre correspondant à la tête, ce noeud étant étiqueté par une catégorie syntaxique pré-terminale (ou *part-of-speech*), comme `v` (verbe) ou `nc` (non commun). L'ancrage d'un arbre τ par une forme f se fait par unification de structures de traits appelées *hypertag* (Kinyon, 2000), avec un hypertag pour τ spécifiant les constructions syntaxiques couvertes par celui-ci et un hypertag pour f indiquant les constructions autorisées pour celle-ci. L'unification des deux hypertags réalise l'ancrage et permet de bloquer certaines traversées dans l'arbre factorisé, évitant en autre des phénomènes de surgénération. Au final, pour être complet et permettre de reconstruire un arbre d'analyse, l'arbre de dérivation doit non seulement se souvenir du nom des arbres mais également des hypertags résultant des ancrages, de manière à pouvoir correctement identifier le parcours suivi dans chaque arbre factorisé.

En sus des noeuds ancre, FRMG introduit aussi des noeuds *co-ancre*. Ces noeuds feuilles sont également étiquetés par des catégories pré-terminales, mais l'opération de co-ancrage n'utilise pas d'hypertag. De tels noeuds sont par exemple utilisés pour des catégories très simples comme les clitiques (comme *il*, figure 5), les prépositions ou certains pronoms. Sans trop entrer dans les détails, un cas plus limite est donné par le traitement de constructions verbales figées comme *prendre conscience* où *prendre* sera l'ancre verbale et *conscience* une co-ancre de catégorie `ncpred`, avec un mécanisme ad-hoc permettant de transférer l'hypertag du nom prédicatif avec l'hypertag sous-spécifié du verbe. Les opérations de co-ancrage laissent une trace dans l'arbre de dérivation en spécifiant non pas un arbre cible mais une forme cible. Les arbres FRMG admettent aussi des noeuds feuilles lexicaux décorés par des formes ou des lemmes (ou des disjonctions de formes ou lemmes). La *lexicalisation* de ces noeuds avec des formes de la chaîne d'entrée se traduit aussi par des traces dans l'arbre de dérivation.

Une autre extension découle du scanner utilisé par FRMG qui permet, à certains endroits, d'ignorer des portions de la chaîne d'entrée considérées comme étant du bruit. Ainsi, le pré-traitement effectué par l'outil SXPipe (Sagot & Boullier, 2008) assigne la catégorie syntaxique `_EPSILON` à des mots comme *euh* ou à des mots répétés comme *le* dans « *le le chat mange* ». L'arbre de dérivation conserve une trace des opérations de sauts, en les affectant de manière arbitraire à l'arbre en cours de reconnaissance pour le pseudo-noeud nommé `skip` (*euh*, figure 5).

Enfin, FRMG étant en fait une TAG avec décorations (F-TAG), les noeuds sont décorés par des paires de structures de traits, celles-ci précisant essentiellement des traits morpho-syntaxiques comme le genre ou le nombre. Sans rentrer dans le détail des opérations TAG, l'arbre de dérivation associe en conséquence une décoration `bot` pour une substitution et une paire de décorations `top` et `bot` pour une adjonction.

Le dernier élément d'information concerne *l'empan* associé à chaque opération, c'est-à-dire la portion de la chaîne d'entrée couverte par l'arbre ou le mot cible. Dans le cadre d'analyse de treillis ambigus de mots représentés par des automates à états finis, ces empans sont généralement donnés par une paire d'états $(\mathrm{start}, \mathrm{end})$ de l'automate et par un quadruplet $(\mathrm{start}, \mathrm{end}, \mathrm{hstart}, \mathrm{hend})$ pour l'adjonction d'un arbre auxiliaire β couvrant $(\mathrm{start}, \mathrm{end})$ avec un noeud pied couvrant un *trou* $(\mathrm{hstart}, \mathrm{hend})$. Pour FRMG, la quasi-totalité des arbres auxiliaires sont en fait utilisables comme des arbres d'insertion (TIG - (Schabes & Waters, 1995)) n'adjoinant du matériel que d'un seul coté du noeud d'adjonction, ce qui conduit, en termes d'empan, à des signatures de la forme $(\mathrm{start}, \mathrm{end}, \mathrm{start}, \mathrm{start})$ ou $(\mathrm{start}, \mathrm{end}, \mathrm{end}, \mathrm{end})$.

En bref, la structure d'arbre de dérivation fournie par FRMG comprend des informations sur une *opération* effectuée sur le noeud source d'un arbre source. Une opération se caractérise par son type (substitution, adjonction, ...), son empan, une ou deux décorations `top` et `bot`, l'arbre ou la forme cible, et l'hypertag de l'arbre cible. Les arbres de dérivation correspondent à des structures concrètes dans le sens où ils essaient de conserver une trace complète de tout ce qui s'est passé, dont de nombreux détails pas nécessairement utiles pour des traitements ultérieurs. [1]

4 Arbres de dépendances

Les arbres de dérivation TAG se convertissent facilement en arbres de dépendances dans le cas de grammaires lexicalisées. L'idée de base est de transformer chaque étape de dérivation (symbolisée par un arc dans l'arbre de dérivation) en une dépendance : une opération de dérivation entre un arbre source τ_1 et un arbre cible τ_2 se traduit par une dépendance entre la tête de τ_1 comme gouverneur et la tête de τ_2 comme gouverné. Ainsi, l'arbre de dérivation de la figure 2b devient l'arbre de dépendance de la figure 3, où l'on voit que le mot *à* n'est pas, par défaut, rattaché car correspondant à un noeud lexical dans l'arbre `tn1pn2` qui n'apparaît pas comme opération dans le modèle de base des arbres de dérivation.

Dans le cadre de FRMG, la conversion est étendue naturellement aux cas où la cible est un noeud co-ancre ou un noeud lexical (traitant ainsi le cas de *à* qui apparaît comme une co-ancre de catégorie `prep` dans l'arbre FRMG correspondant à `tn1pn2`, donnant lieu à la dépendance en pointillé). Les labels des arcs sont donnés par les labels des noeuds sources. Pour les noeuds feuilles de substitution ou de co-ancrage, le label utilisé dans FRMG traduit en général leur fonction syntaxique comme `sujet`, `objet` ou `xcomp` (pour les arguments phrastiques). Pour les noeuds internes sur lesquels sont appliquées des adjonctions, le label

1. Cependant, certains « détails » sont parfois utiles pour certaines applications, par exemple le mode des verbes ou le caractère saturé ou non d'un groupe nominal (par un déterminant).

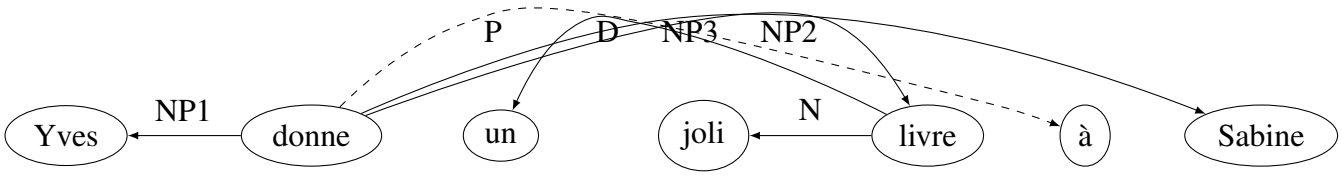


FIGURE 3 – arbre simple de dépendance

correspond généralement à la catégorie syntaxique du noeud (plus un index) et n'est malheureusement pas toujours très informatif.

Certains arbres FRMG ne possèdent pas de noeud ancre, conduisant à introduire une pseudo-ancre vide de catégorie syntaxique X positionnée en start pour un tel arbre τ non ancré, de racine X, et couvrant un empan démarrant à start.

Par construction, l'ensemble des adjonctions portant sur n'importe quel noeud N d'un arbre τ sont vues comme des dépendances ayant comme gouverneur l'ancre lexicale A de τ. En pratique, ce comportement n'est pas toujours intuitif car on aimerait parfois prendre une valeur lexicale dominée par N comme gouverneur. Ainsi, dans le cadre de la phrase « *il a pris pleine conscience du problème* », la théorie stipule une dépendance *pris* $\longrightarrow$ *pleine* mais l'intuition une dépendance *conscience* $\longrightarrow$ *pleine*. Il a donc été décidé qu'une adjonction portant sur un noeud co-ancre est transformée en une dépendance vers celui-ci, tout en se rappelant le noeud ancre. [2]

Enfin, rappelons que, du fait des adjonctions, les structures de dépendance obtenus ne sont pas nécessairement projectives. Il est aussi à noter que l'utilisation d'arbres TIG conduit à la possibilité d'adjonctions multiples sur un noeud en place des adjonctions chaînées dans le cas des TAG (Schabes & Shieber, 1994) [3]. En terme de dépendance, on obtient alors directement des dépendances multiples sur le bon gouverneur (par exemple pour des adjectifs comme dans *le joli petit chat noir*) plutôt que sur les ancres des arbres auxiliaires intermédiaires (chaînage d'adjectifs).

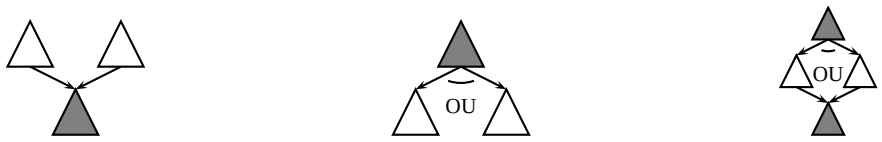
5 Forêts de dépendances

FRMG retourne l'ensemble de toutes les dérivations possibles pour une phrase, en s'appuyant sur un algorithme d'analyse tabulaire à la Earley conservant des traces de calculs sous forme d'*items* pour réaliser du partage de calculs. L'ensemble des arbres de dérivation est retourné sous forme d'une forêt partagée, extraite de ces items grâce à des pointeurs arrière permettant de remonter des items à leurs parents (Billot & Lang, 1989). En oubliant les décorations, une telle forêt est de taille polynomiale en la longueur n de la phrase ($O(n^6)$ en étant efficace) mais peut représenter un nombre exponentiel (voire infini) d'arbres. La figure 4 schématise les deux types de partage présents dans ces forêts, à savoir le partage de sous-arbres et le partage de contextes grâce à des graphes ET-OU.

Concrètement, le partage par sous-arbre dans les arbres de dérivation se traduit par le fait qu'une même opération comme la substitution d'un arbre cible τ couvrant un certain empan s peut s'effectuer sur dif-

2. On peut envisager d'étendre ce mécanisme pour d'autres cas par un système de règles. Par contre, il paraît plus difficile de gérer des changements d'orientation des dépendances, parfois souhaités, par exemple pour des adjonctions prédicatives.

3. L'ordre des adjonctions est fourni par les empans, ce qui est potentiellement incomplet en cas d'empan vide.



(a) par sous-arbre (b) par contexte (c) par sous-arbre et contexte

FIGURE 4 – Type de partage d'arbres

férents noeuds sources. Le partage par contexte se traduit par le fait que, pour un certain arbre source couvrant un certain empan, plusieurs ensembles alternatifs d'opérations sont possibles. La conversion en forêt partagée de dépendances traite ces deux cas. Le partage de contextes se traduit par le fait qu'un gouverneur (ancre d'un arbre source) peut être associé à plusieurs ensembles alternatifs de dépendances, ces ensembles pouvant partager des dépendances. Le partage de sous-arbres se traduit par plusieurs gouverneurs en compétition pour un gouverné. Ces éléments se retrouvent dans la figure 5 qui donne une forme visuelle de la forêt de dépendance obtenue par conversion pour la phrase « *il observe euh une maman [avec ses jumelles]*$_{\text{PP}}$ » avec une ambiguïté de rattachement sur *maman* ou *observe* pour le groupe prépositionnel PP et des ambiguïtés lexicales sur *jumelles* (incluant la transcatégorisation d'un adjectif).

La figure 5 correspond à une approximation du format **DEP XML** de représentation des forêts partagées de dépendance, format XML qui comprend :

– des éléments **cluster** (matérialisés par les rectangles extérieurs) associés aux mots de la phrase et caractérisés par une paire de positions `left` et `right` dans l'automate. Du fait des ambiguïtés de segmentation provenant de SXPipe, en particulier sur les mots composés, un mot simple (*token*) peut se retrouver dans plusieurs clusters.
– des éléments **node** (matérialisés par les rectangles intérieurs) pointant vers un `cluster` et décorés d'une forme `form` (rectifiée en cas de correction orthographique fournie par SXPipe), d'un lemme `lemma`, d'une catégorie syntaxique `cat`, un arbre ancré `tree` de catégorie maximale `xcat` et d'un ensemble de dérivations `deriv`. Plusieurs nœuds peuvent être associés à un même cluster.
– des éléments **op** (non matérialisés), historiquement traces des opérations dans la forêt de dérivation mais intuitivement interprétables comme des constituants. Ils portent une catégorie syntaxique non-terminale `cat` comme `S`, un empan `span` simple ou à trou, et incluent une ou deux décorations `top` ou `bot` introduites par les éléments **narg**. Un op O est aussi associé à un ensemble de dérivations `deriv`, en compétition pour construire O. Chaque dérivation étant aussi associée à un noeud gouverneur, on constate donc que O peut avoir plusieurs noeuds tête en compétition.
– des éléments **edges** (matérialisés par les arcs) mettant en relation un nœud `source` gouverneur s avec un nœud `target` gouverné t, également décorés par un `label` et un `type` (substitution, adjonction, ...). Plus finement, un arc e est utilisé (*traversé*) par un sous-ensemble $\mathcal{D}_e$ des dérivations $\mathcal{D}_s$ de son nœud source s avec $\mathcal{D}_e$ « partitionné » entre plusieurs sous-éléments **deriv** de e, chacun portant, en attribut `names`, un sous-ensemble $\mathcal{D} \subset \mathcal{D}_e$ et permettant de relier un constituant source `source_op` (ancré par s) avec un constituant cible `target_op` (ancré par t) dont l'empan est donné par `span`. Pour traiter le cas particulier des affectations de dépendances sur les co-ancres, l'attribut `reroot_source` est utilisable sur **deriv** pour conserver le lien vers le noeud ancre.
– des éléments **hypertags** (non matérialisés) portant des dérivations et incluant une structure de traits. Un hypertag est associable, au travers des dérivations, à plusieurs constituants et noeuds tête en compétition.

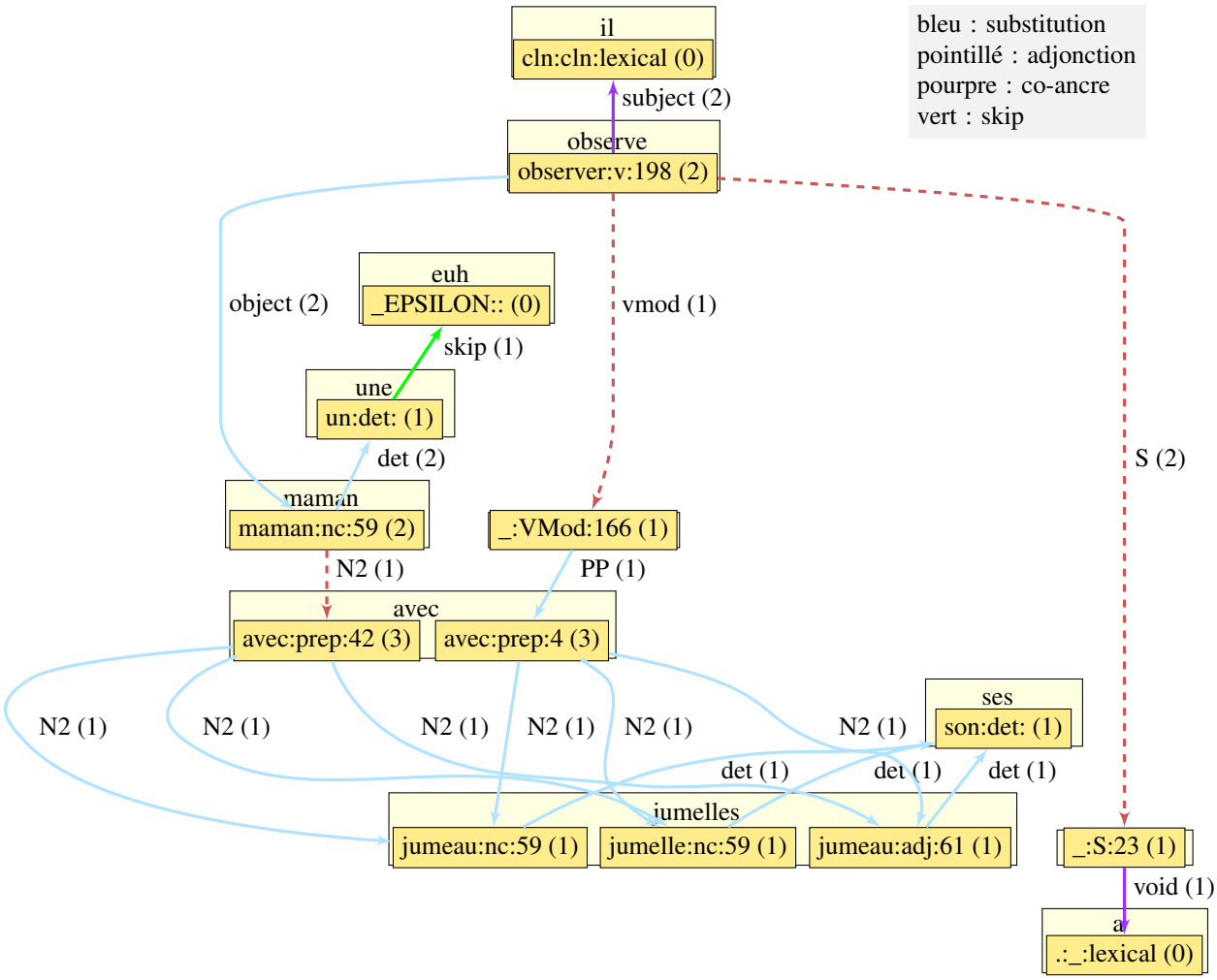


FIGURE 5 – Forêt partagée de dépendances pour « *il observe euh une maman avec ses jumelles.* »

Les pseudo-ancres mentionnées précédemment donnent lieu à des éléments **cluster** lexicalement vides associés à des éléments **node** également lexicalement vides (cas de `VMod:165` et de `S:23` dans la figure 5), mais néanmoins porteurs d'un arbre, d'une catégorie maximale et d'un ensemble de dérivations.

Le format DEP XML ne matérialise pas explicitement les dérivations (en tant qu'éléments) mais celles-ci forment le liant de l'ensemble des composants (nœuds, arcs, op et hypertags). Il est aussi intéressant de remarquer que les forêts de dépendances partagent moins bien que les forêts de dérivations. En effet, la conversion tend à *ouvrir* la forêt de dérivation du fait qu'un constituant (symbolisé par un élément **op**) peut avoir plusieurs noeud têtes. Ainsi, supposons un arbre τ ancré par h sur lequel les opérations $op_1, \ldots, op_m$ ont été effectuées (correspondant à une dérivation de τ), chaque opération op_i étant associées à n_i têtes $h_{i,j}$ différentes. Alors, nous aurons toutes les dépendances $h \longrightarrow h_{ij}$ regroupables en $n_1.\cdots.n_m$ dérivations potentielles pour h. Le nombre de dérivations associées à un noeud gouverneur peut donc croître de manière exponentielle. Ceci est particulièrement vrai suite à l'utilisation de l'adjonction multiple des TIG qui peut fait croître le paramètre m. De longues séquences d'attachements prépositionnels sont ainsi parfois susceptibles de conduire à de très grosses forêts de dépendances.

La vue graphique fournie par la figure 5 permet de facilement repérer les points d'ambiguité, sur *avec* (en relation avec l'attachement sur *maman* ou *observe*) et sur *jumelles*. Le niveau d'ambiguité est précisé grâce

aux nombres entre parenthèses portés par les noeuds et les dépendances. Sur un noeud, ce nombre indique combien de dérivations partent de celui-ci en tant que noeud gouverneur. Sur une dépendance d, il indique combien de dérivations du gouverneur de d transitent par d. Ainsi, 2 dérivations partent de *observe*, les 2 deux utilisant les arcs `object` et `subject`, mais une seule utilisant l'arc `vmod`.

Ces nombres permettent d'estimer le taux d'ambiguïté d'une phrase ramené au niveau du mot. La première mesure est le taux d'ambiguïté moyen par mot défini par $\alpha = \frac{1+\#\text{edges}}{\#\text{clusters}} - 1$. Ce taux indique le nombre moyen d'arcs entrants en surnombre par mot, en considérant qu'en cas de non-ambiguïté, on a un et un seul arc entrant par mot d'où $\alpha = 0$. Un taux de 1 indique 2 arcs entrants en moyenne par mots, donnant normalement de l'ordre de 2^n analyses pour une phrase de longueur n. Néanmoins, cette mesure très simple qui reflète la structure graphique des dépendances ne prend pas en compte les interactions entre arcs données par les dérivations et a tendance à sous-estimer le niveau réel d'ambiguïté. Une mesure alternative est donnée par le nombre moyen de dérivations par mot $\beta = \frac{\Sigma_{n \in \text{nodes}} |\text{deriv}_n|}{\#\text{nodes}}$. Pour la figure 5, nous obtenons ainsi $\alpha = 0,7$ et $\beta = 1,2$.

6 Désambiguisation

Dans un contexte applicatif, il est souvent nécessaire de ne considérer qu'une analyse par phrase (si possible la bonne !). Un désambiguisateur, écrit en DYALOG, existe et s'appuie sur un algorithme en programmation dynamique de recherche de l'analyse A de plus fort poids donné par la somme des poids des arcs (et, dans une bien moindre mesure, des nœuds) participant à A. Le poids d'un arc s'exprime comme somme des poids donnés par des règles élémentaires exprimées sous forme de motifs prenant en compte les composant de l'arc courant (nœuds source et cible, type, label) et éventuellement des arcs frères, fils ou parents, voire des arcs en compétition. Les poids sont choisis de manière heuristique et donnent, pour l'instant, de meilleurs résultats que ceux obtenus par apprentissage. À ce jour, le désambiguisateur comporte environ 150 règles élémentaires. Par exemple, citons l'existence de règles favorisant les arcs remplissant la valence d'un verbe (sujet, objet, ...), la présence d'un sujet avant son verbe, l'inversion du sujet si certaines conditions sont remplies, etc. D'autres règles pénalisent les dépendances à longue distance, les transcatégorisations non nécessaires, certaines constructions possibles mais improbables, etc.

```
edge_cost_elem( Name::'+ATTR',
    edge{ label => comp, source => node{ cat => v}, target => node{ cat => adj } },
    100 ).
```

La notion de dépendance se prête naturellement bien à la définition de ces règles de désambiguisation (comme la règle ci-dessus favorisant légèrement (+100) les attributs adjectivaux par rapport à d'autres catégories syntaxique). Elle permet aussi de prendre en compte des préférences bi-lexicalisées et même tri-lexicalisées pour des restrictions de sélection (obtenues sur comptage de dépendances sur de larges corpus). Par contre, l'algorithme de désambiguisation travaille sur une approximation de la forêt dans le sens où les poids sont calculés sur les dépendances indépendemment des dérivations associées. En toute logique, ce n'est pas correct dans le cas de règles consultant une dépendance mais aussi des dépendances soeurs, parentes ou filles, éventuellement non présentes pour certaines dérivations. Néanmoins, le coût de l'algorithme (table 1), est déjà relativement élevé et parfois très long sur certaines phrases très ambiguës.

L'algorithme de désambiguisation retourne un arbre de dépendance toujours représentable en format DEP XML. Il est également possible de convertir le résultat dans les formats EASy ou PASSAGE (Paroubek

et al., 2009) utilisés par les campagnes d'évaluation syntaxique du français.

7 Évaluation

La table 1 donne les valeurs moyennes, médianes, et maximales des mesures d'ambiguïté α et β sur environ 4000 phrases du corpus de référence EASy (Paroubek *et al.*, 2008) couvrant divers styles (littéraire, journalistique, médical, oral, ...). On observe un net décalage entre les évolutions de α et β. Ces valeurs peuvent aussi être contrastées avec les distributions correspondantes pour les longueurs des phrases et les temps d'analyse et de désambiguisation (avec un timout de 110s). La table 2, calculée sur 2,9 millions de phrases de Wikisource avec des analyses complètes et des longueurs n entre 1 et 231 (moyenne de $12,74$), suggère que la mesure β est légèrement mieux corrélée pour les temps d'analyse, de conversion vers DEP XML et de désambiguisation (vers Passage), avec une excellente corrélation pour DEP XML ($R^2 = 0,91$). Notons également que la complexité de l'analyse par rapport à la longueur n semble en moyenne moindre que $O(n^2)$ et celle de la désambiguisation moindre que linéaire, ce qui semble indiquer que le coût élevé de la désambiguisation (table 1) est du à un mauvais facteur constant peut-être réductible.

type	n	avg	median	%90	%99	max
longueur	3879	19,30	16,00	40,00	69,00	136,00
α	3869	1,06	0,90	2,00	4,00	9,10
β	3869	7,32	2,30	11,90	101,00	967,40
t. analyse (s)	3879	0,58	0,14	0,95	5,26	110,00
t. désamb (s)	3869	0,92	0,24	1,23	11,35	110,00

TABLE 1 – Quelques statistiques sur les niveaux d'ambiguité et les temps d'analyse et de désambiguisation

	analyse	xmldep	desamb
n^a	R^2=0,77 a=1,61	R^2=0,77 a=1,10	R^2=0,62 a=0,80
$(\Sigma_n \|\text{deriv}_n\|)^a$	R^2=0,76 a=0,81	R^2=0,91 a=0,60	R^2=0,85 a=0,47
$n^a(1+\alpha)^b$	R^2=0,80 a=1,46 b=0,91	R^2=0,90 a=0,92 b=1,12	R^2=0,76 a=0,64 b=0,94
$n^a\beta^b$	R^2=0,82 a=1,19 b=0,54	R^2=0,91 a=0,68 b=0,54	R^2=0,83 a=0,37 b=0,55

TABLE 2 – Indication de corrélation (sur 2.9Mphrases de Wikisource)

8 Conclusion

Nous avons repris et approfondi les relations existant entre arbres de dérivations TAG et arbres de dépendances, en les transposant en particulier au niveau de forêts partagées. Il en résulte une description associée à une représentation XML qui permet la conception d'applications exploitant des structures de dépendances riches en information. La représentation proposée s'applique par construction aux forêts de dépendances issue d'analyses TAG, mais est aussi utilisable pour de simples arbres après désambiguisation. Nous pensons qu'elle est adaptable à toute sorte de forêts de dépendances, permettant de combiner à la fois des informations strictement de dépendance mais aussi des informations de constituance. Ce choix va dans le sens d'autres formats émergeants comme TIGER (Brants & Hansen, 2002) ou la proposition de norme ISO SynAF (Declerck, 2008), qui, eux, ne prennent cependant pas en compte la notion d'ambiguité.

Références

BILLOT S. & LANG B. (1989). The structure of shared forests in ambiguous parsing. In *Proc. of the 27 Annual Meeting of the Association for Computational Linguistics*.

BRANTS S. & HANSEN S. (2002). Developments in the TIGER annotation scheme and their realization in the corpus. In *Proceedings of the Third Conference on Language Resources and Evaluation (LREC 2002)*, p. 1643–1649, Las Palmas.

CANDITO M.-H. & KAHANE S. (1998). Can the derivation tree represent a semantic graph ? an answer in the light of meaning-text theory. In *In Proc. of TAG+4*, Philadelphia.

DECLERCK T. (2008). A framework for standardized syntactic annotation. In *Proceedings of the Sixth Conference on International Language Resources and Evaluation* : European Language Resources Association (ELRA) ELRA/ELDA.

FERNANDEZ M., VILLEMONTE DE LA CLERGERIE E. & VILARES M. (2007). From text to knowledge. In *Proc. of EUROCAST'07 (Eleven international conference on Computer Aided Systems theory)*.

GARDENT C. & KALLMEYER L. (2003). Semantic construction in feature-based tree-adjoining grammars. In *Proc. of the 10th Conference of the European Chapter of ACL*.

JOSHI A. & RAMBOW O. (2003). A formalism for dependency grammar based on tree adjoining grammar. In *Proceedings of the Conference on Meaning-Text Theory (MTT)*, Paris.

JOSHI A. K. (1987). An introduction to tree adjoining grammars. In A. MANASTER-RAMER, Ed., *Mathematics of Language*, p. 87–115. Amsterdam/Philadelphia : John Benjamins Publishing Co.

KINYON A. (2000). Hypertags. In *Proc. of COLING*, p. 446–452.

PAROUBEK P., ROBBA I., VILNAT A. & AYACHE C. (2008). Easy, evaluation of parsers of french : what are the results ? In *Proceedings of the 6th International Conference on Language Resources and Evaluation (LREC)*, Marrakech, Morroco.

PAROUBEK P., VILLEMONTE DE LA CLERGERIE É., LOISEAU S., VILNAT A. & FRANCOPOULO G. (2009). The PASSAGE syntactic representation. In *7th International Workshop on Treebanks and Linguistic Theories (TLT7)*, Groningen.

SAGOT B. & BOULLIER P. (2008). SxPipe 2 : architecture pour le traitement pré-syntaxique de corpus bruts. *Traitement Automatique des Langues*, **49**(2).

SCHABES Y. & SHIEBER S. M. (1994). An alternative conception of tree-adjoining derivation. *Computational Linguistics*, **20**(1), 91–124.

SCHABES Y. & WATERS R. C. (1995). Tree insertion grammar : a cubic-time, parsable formalism that lexicalizes context-free grammar without changing the trees produced. *Fuzzy Sets Syst.*, **76**(3), 309–317.

THOMASSET F. & VILLEMONTE DE LA CLERGERIE E. (2005). Comment obtenir plus des méta-grammaires. In *Proceedings of TALN'05*, Dourdan, France : ATALA.

VILLEMONTE DE LA CLERGERIE E. (2002). Construire des analyseurs avec DyALog. In *Proc. of TALN'02*.

VILLEMONTE DE LA CLERGERIE E. (2010). Building factorized tags with meta-grammars. In *Proc. of TAG+10*, Yale University. To appear.

ÉRIC DE LA CLERGERIE, SAGOT B., STERN R., DENIS P., RECOURCÉ G. & MIGNOT V. (2009). Extracting and visualizing quotations from news wires. In *Proceedings of LTC 2009*, Poznan, Pologne.